Todd Wenker

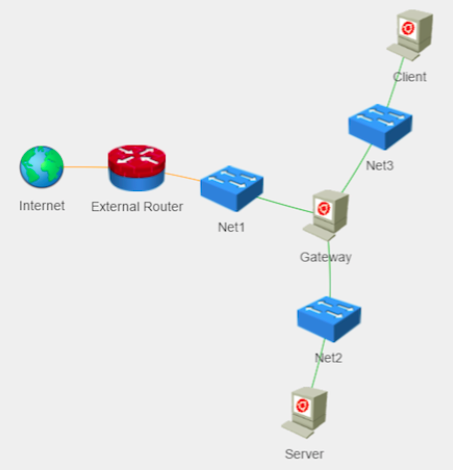
ASU ID: 1206233882

Lab Assignment 4

DNS Local Attacks

**Summary:** The goal of the lab is to conduct three different attacks by manipulating DNS servers and their replies. By providing a host with the incorrect IP address in response to a DNS request, the host can be led to unsafe sites.

**Network Setup:**



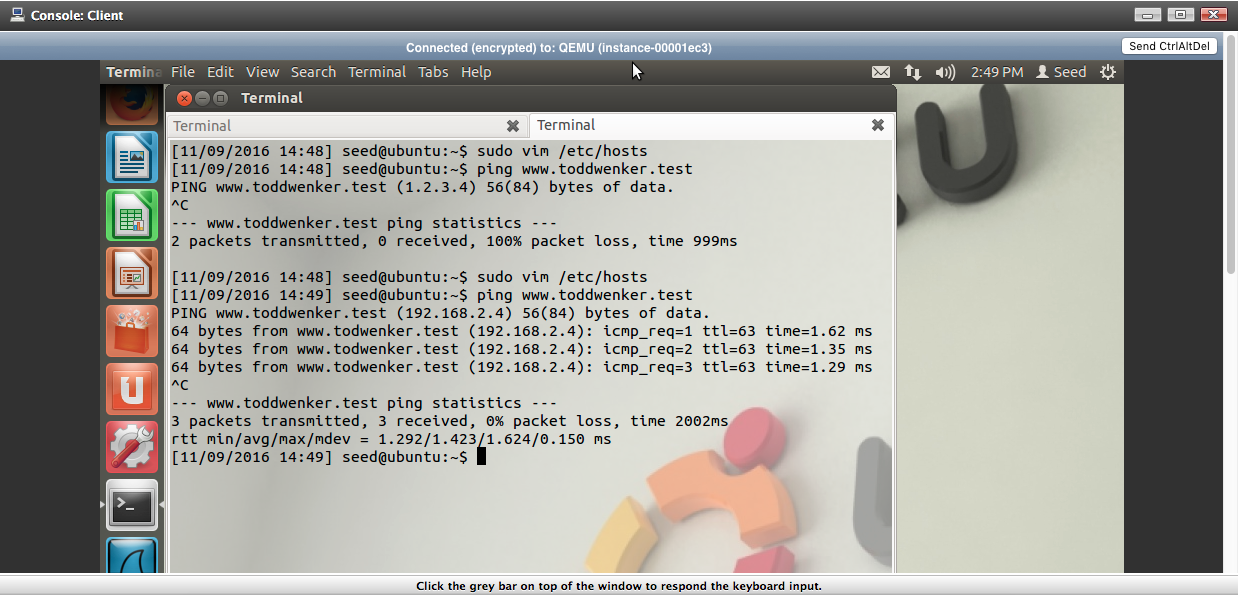
The network is setup with three hosts: Gateway, Server, and Client. Gateway is set up as a DNS server for the subnet containing Gateway and Server hosts.

**Software Packages Used:**

* Wireshark
* Netwox
* Bind9

**Task 1 --- DNS Attack on a Compromised Machine**

Assuming that we already have access to a machine, manipulating DNS responses is straightforward. This is because hosts will search their own files before sending DNS requests to name servers. To add a static entry to the host’s list of known domain names, the /etc/hosts file needs to be edited. By adding the line *1.2.3.4 www.toddwenker.test,*the host will assume that the domain name www.toddwenker.test maps to the IP address 1.2.3.4. By using the command *ping www.toddwenker.test*, the machine will attempt to contact the faulty IP address, shown below.

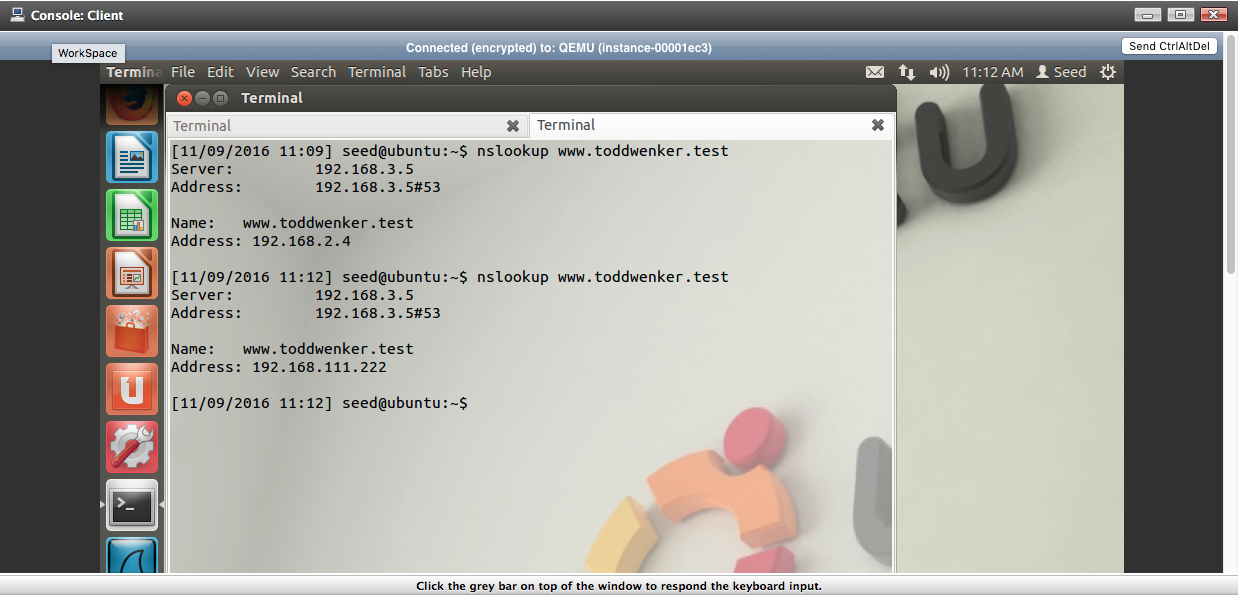


After the first ping, the host assumes 1.2.3.4 is the IP address and is sending packets to it.

The second *ping www.toddwenker.test* was done after the /etc/hosts file was changed back to its original state and the host is now sending packets to 192.168.2.4, the correct address.

**Task 2 --- Directly Spoof DNS Response**

Even without compromising a machine, DNS responses can still be spoofed to trick a host into translating a domain name into the incorrect IP address. If a packet can be made using spoofed port, IP address, UDP checksum, and transaction IP numbers, a host can be sent a packet that it thinks is from the appropriate name server in response to their DNS request when it actually contains incorrect information. The command *sudo netwox 105 --hostname “www.toddwenker.test” --hostnameip “192.168.111.222” –authns “gateway.toddwenker.test” --authnsip “192.168.3.5” --device “eth0”* is used to spoof a DNS response. The netwox suite of programs is used, with 105 being the program that spoofs DNS responses. The ‘hostname’ is the domain name that netwox is waiting to see requested on the targeted name server, the ‘hostnameip’ is the IP address that will be sent to a host that requests the ‘hostname’ IP address. The ‘authns’ is the domain name of the name server being targeted and ‘authnsip’ is the IP address of said name server. Finally, the ‘device’ is the Ethernet connection where the message is being sent. Below is the response of the command *nslookup “www.toddwenker.test”. t*

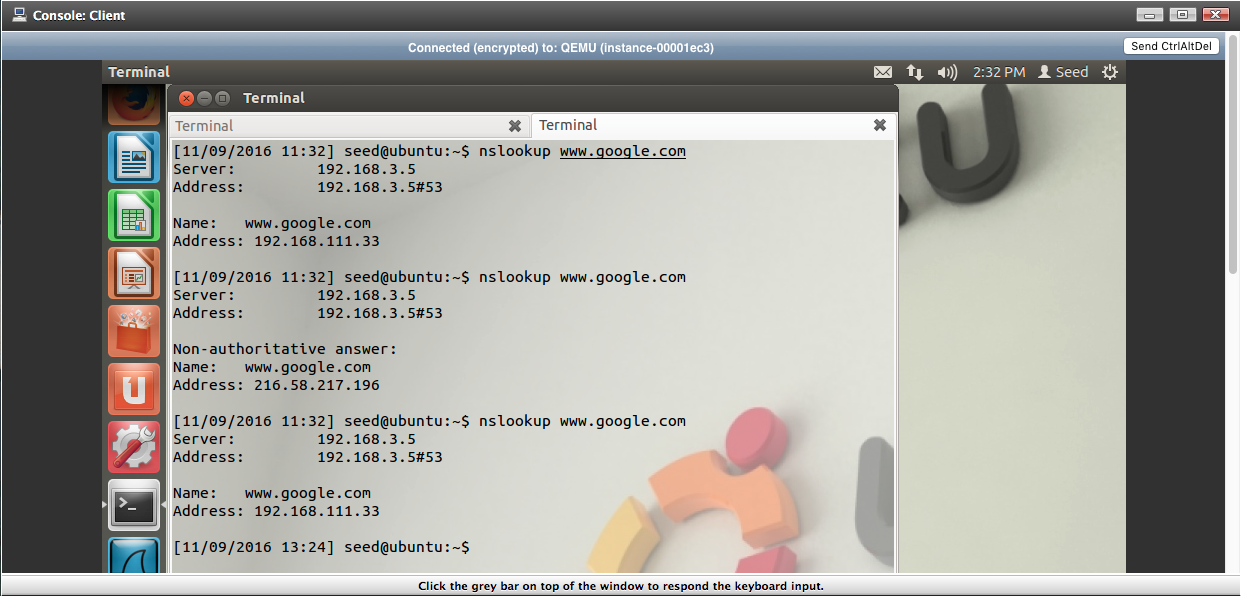


The *nslookup www.toddwenker.test* was run twice, due to the first attempt returning the correct IP address of 192.168.2.4. This is due to the valid DNS response from the name server reaching the Client machine before the spoofed response. The second attempt proved to be successful, as the returned IP address was the spoofed IP address of 192.168.111.222.

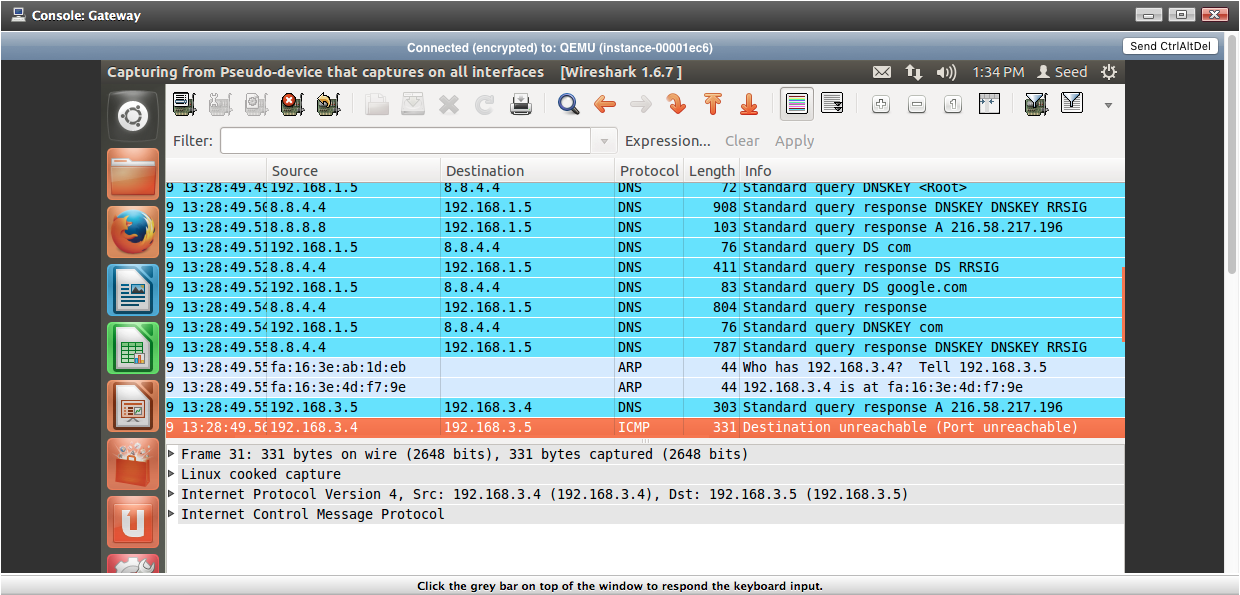
**Task 3 --- DNS Server Cache Poisoning**

The next attack is the DNS server cache poisoning attack. When a name server receives a DNS request, the name server first checks the local cache to see if the requested domain name’s IP address is there. If the domain name is not there, the server will then query another name server to find the domain name. The goal of the attack is to enter the incorrect IP address for a specific domain name into the cache of the targeted name server. That way, when other hosts query the server, the server will respond with the incorrect IP address. This means that the attacking machine does not need to match each message sent out of the name server with a spoofed message as the name server itself should be sending the incorrect messages.

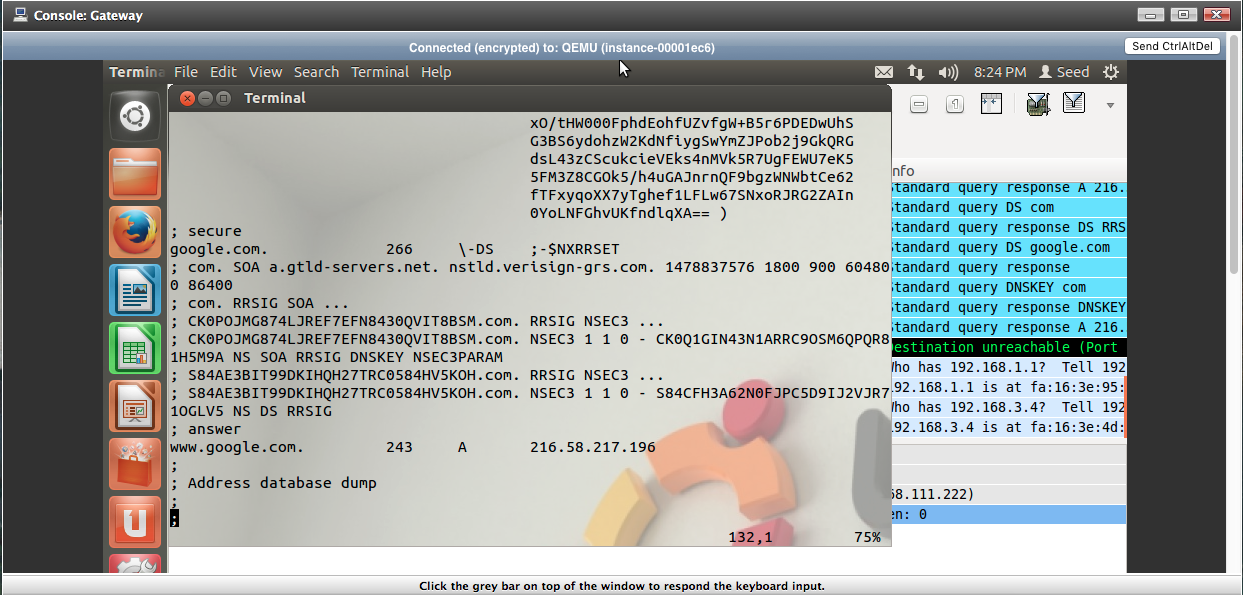
First, the cache on the name server needs to be flushed so that it can store the incorrect IP address. This is done with the command *sudo rndc flush* on the Gateway machine*.* The ‘rndc’ stands for Remote Name Daemon Control, which allows the administration of the daemon responsible for caching domain names. Next, the command *sudo netwox 105 --hostname “www.google.com” --hostnameip “192.168.111.33” –authns “gateway.toddwenker.test” --authnsip “192.168.3.5” --device “eth0” --ttl 600 --spoofip “raw”* is used from the Client machine so as when the Gateway machine receives a DNS request for the domain name ‘www.google.com’, it will receive a spoofed message that it believes is from the name server containing the correct IP address to ‘www.google.com’. This spoofed response will be stored in the cache before being sent to the Client machine. Below are the results of the attack.

****

Above is the result of the command *nslookup www.google.com* from the Client machine. Despite successfully receiving the fake IP address, the second attempt returned the true IP address instead.



Above is the Wireshark capture on the Gateway machine, started before the *netwox* command was used. Despite the attempt to poison the Gateway’s cache, it still ended up querying the DNS name server 8.8.4.4.



Above is the DNS cache dump of the Gateway machine created by using the command *sudo rndc dumpdb -cache* to create the file and using the command *vim /var/cache/bind/named\_dump.db* to view the created file. Noticeably, the domain name ‘www.google.com’ is correlated to the IP address 216.58.217.196, which is the correct address but not what it should be if the cache was poisoned. As such, the attack failed as the Gateway machine’s cache contains the correct IP address.

**Conclusion**

While DNS based attacks are not capable of compromising a machine outright, what they are capable of doing is sending victims to incorrect IP addresses. If a host is already compromised, the attacker can edit the victim’s hosts file to create a static connection between a specific domain name and an IP address. If a machine is not compromised, the attacker can instead create spoofed wait till a victim sends out a DNS request and create a spoofed answer so the victim then tries to connect to the incorrect IP address. The downside of this method is that the response sent by the name server could potentially reach the victim before the attacker’s spoofed response, nullifying the attack. A way to circumvent this is to instead poison the DNS cache of the name server. By sending a spoofed DNS response message to the name server, the requested domain name will be associated with the incorrect IP address. This way, whenever that name server is queried for that domain name, the server itself will generate the DNS response with the faulty IP address.